

Full step Rotary Encoder

1.0.1

Generated by Doxygen 1.8.14

Contents

1	3 speed Rotary Encoder Full Step library for Arduino	1
2	Class Index	5
2.1	Class List	5
3	File Index	7
3.1	File List	7
4	Class Documentation	9
4.1	RotaryFullStep Class Reference	9
4.1.1	Detailed Description	9
4.1.2	Constructor & Destructor Documentation	9
4.1.2.1	RotaryFullStep()	9
4.1.3	Member Function Documentation	10
4.1.3.1	read()	10
5	File Documentation	11
5.1	src/RotaryFullStep.cpp File Reference	11
5.1.1	Detailed Description	11
5.2	src/RotaryFullStep.h File Reference	11
5.2.1	Detailed Description	12
	Index	13

Chapter 1

3 speed Rotary Encoder Full Step library for Arduino

This is an optimized three speed Rotary Encoder library for Arduino which supports:

- Full step Rotary Encoder types.
- Detect three rotation speeds.
- Configurable sensitivity.
- Polling and interrupts.
- Optional button.

Hardware

Connect the two rotary pins to the DIGITAL pins of an Arduino board.

A third rotary button pin is not used in the Rotary library, but can be used in the sketch.

Tested with Arduino IDE v1.8.5 on hardware:

- Arduino UNO
- Arduino Nano
- Arduino Micro
- Arduino Pro or Pro Mini
- Arduino Mega or Mega2560
- Arduino Leonardo
- WeMos D1 R2 & mini

Interrupts

Both rotary pins must be connected to a DIGITAL pin with interrupt support, such as `INT0` or `INT1`. This is chip specific. Please refer to the documentation of your board or `attachInterrupt()`.

Arduino UNO example

The connection below can be used for polled and interrupts. An optional button pin can be connected to DIGITAL pin 4.

Examples

The following examples are available:

- Rotary | Interrupt | [InterruptFullStepBasic](#)
- Rotary | Interrupt | [InterruptFullStepButton](#)
- Rotary | Interrupt | [InterruptFullStepCounter](#)
- Rotary | Polled | [PolledFullStepBasic](#)
- Rotary | Polled | [PolledFullStepButton](#)
- Rotary | Polled | [PolledFullStepCounter](#)

Usage

Read rotary with polling

```
{c++}
#include <RotaryFullStep.h>

// Configure rotary pins connected to your Arduino board
#define ROTARY_PIN1 2
#define ROTARY_PIN2 3

// Initialize full step rotary encoder, default pull-up enabled, default
// sensitive=100
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2);

// Or initialize full step rotary encoder, pull-up disabled, default sensitive=100
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2, false);

// Or initialize full step rotary encoder, pull-up enabled, sensitive 1..255
// A higher value is more sensitive
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2, true, 150);

void loop()
{
    int rotaryState = rotary.read();

    // rotaryState = -3: Turn left fastest
    // rotaryState = -2: Turn left faster
    // rotaryState = -1: Turn left
    // rotaryState = 0: No change
    // rotaryState = 1: Turn right
    // rotaryState = 2: Turn right faster
    // rotaryState = 3: Turn right fastest
}
```

Read rotary with interrupts

```

{c++}
#include <RotaryFullStep.h>

// Connect rotary to Arduino DIGITAL pins with interrupt support:
//
// +-----+-----+-----+
// |           Board           | DIGITAL interrupt pins |
// +-----+-----+-----+
// | Uno, Nano, Mini, other 328-based | 2, 3 |
// | Mega, Mega2560, MegaADK         | 2, 3, 18, 19, 20, 21 |
// | Micro, Leonardo, other 32u4-based | 0, 1, 2, 3, 7 |
// +-----+-----+-----+
//
#define ROTARY_PIN1 2
#define ROTARY_PIN2 3

// Initialize full step rotary encoder, default pull-up enabled, default
// sensitive=100
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2);

// Or initialize full step rotary encoder, pull-up disabled, default sensitive=100
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2, false);

// Or initialize full step rotary encoder, pull-up enabled, sensitive 1..255
// A higher value is more sensitive
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2, true, 150);

void setup()
{
    // Initialize pin change interrupt on both rotary encoder pins
    attachInterrupt(digitalPinToInterrupt(ROTARY_PIN1), rotaryInterrupt, CHANGE);
    attachInterrupt(digitalPinToInterrupt(ROTARY_PIN2), rotaryInterrupt, CHANGE);
}

void rotaryInterrupt()
{
    int rotaryState = rotary.read();

    // rotaryState = -3: Turn left fastest
    // rotaryState = -2: Turn left faster
    // rotaryState = -1: Turn left
    // rotaryState = 0: No change
    // rotaryState = 1: Turn right
    // rotaryState = 2: Turn right faster
    // rotaryState = 3: Turn right fastest
}

```

Installation with Git

Install Git client for Windows

Install a [Git client for Windows](#).

Install Git client for Linux

Open a command prompt and install a Git client for Linux, such as Debian Ubuntu:

```
sudo apt-get install git
```

Windows and Linux

The library must be installed in the Sketchbook directory which is configured in the Preferences dialog box.

1. Click File | Preferences | Settings tab and copy the Sketchbook location. The path on Windows is something like: C:\Users\User\Documents\Arduino The path on Linux is something like↵
: /home/user/Arduino

2. Open a command prompt and type:

```
# Run on Windows:
cd C:\Users\User\Documents\Arduino
# Or run on Linux:
cd ~/Arduino

# Run the git clone library once:
git clone git clone https://github.com/Erriez/ErriezRotaryEncoderFullStep.git

# Update the library:
git pull
```

1. Restart the Arduino IDE.

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

RotaryFullStep	
Full step Rotary Encoder class	9

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

src/ RotaryFullStep.cpp	
Three speed full step Rotary Encoder library for Arduino	11
src/ RotaryFullStep.h	
Three speed full step Rotary Encoder library for Arduino	11

Chapter 4

Class Documentation

4.1 RotaryFullStep Class Reference

Full step Rotary Encoder class.

```
#include <RotaryFullStep.h>
```

Public Member Functions

- [RotaryFullStep](#) (uint8_t pin1, uint8_t pin2, bool pullUp=true, uint8_t sensitive=100)
Constructor full step Rotary Encoder.
- int [read](#) ()
Read Rotary Encoder state.

4.1.1 Detailed Description

Full step Rotary Encoder class.

Definition at line 42 of file RotaryFullStep.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 RotaryFullStep()

```
RotaryFullStep::RotaryFullStep (  
    uint8_t pin1,  
    uint8_t pin2,  
    bool pullUp = true,  
    uint8_t sensitive = 100 )
```

Constructor full step Rotary Encoder.

Parameters

<i>pin1</i>	Rotary Encoder pin 1
<i>pin2</i>	Rotary Encoder pin 2
<i>pullUp</i>	Enable/disable Rotary Encoder pull-up pins
<i>sensitive</i>	Sensitivity value 1..255 (Higher is more sensitive)

Definition at line 83 of file RotaryFullStep.cpp.

4.1.3 Member Function Documentation

4.1.3.1 read()

```
int RotaryFullStep::read ( )
```

Read Rotary Encoder state.

Call this function as fast as possible to prevent missing pin changes. This seems to work for most rotary encoders $\leq 10\text{ms}$.

Returns

Rotary speed and direction -2: Counter clockwise fastest -1: Counter clockwise faster -0: No change 1: Clockwise faster 2: Clockwise fastest

Definition at line 107 of file RotaryFullStep.cpp.

The documentation for this class was generated from the following files:

- [src/RotaryFullStep.h](#)
- [src/RotaryFullStep.cpp](#)

Chapter 5

File Documentation

5.1 src/RotaryFullStep.cpp File Reference

Three speed full step Rotary Encoder library for Arduino.

```
#include <avr/pgmspace.h>
#include "RotaryFullStep.h"
```

Macros

- `#define DIR_NONE 0x00`
- `#define DIR_CW 0x10`
- `#define DIR_CCW 0x20`
- `#define RFS_START 0x00`
- `#define RFS_CW_FINAL 0x01`
- `#define RFS_CW_BEGIN 0x02`
- `#define RFS_CW_NEXT 0x03`
- `#define RFS_CCW_BEGIN 0x04`
- `#define RFS_CCW_FINAL 0x05`
- `#define RFS_CCW_NEXT 0x06`

5.1.1 Detailed Description

Three speed full step Rotary Encoder library for Arduino.

[RotaryFullStep.cpp](#)

Source: <https://github.com/Erriez/ErriezRotaryEncoderFullStep>

5.2 src/RotaryFullStep.h File Reference

Three speed full step Rotary Encoder library for Arduino.

```
#include <Arduino.h>
```

Classes

- class [RotaryFullStep](#)
Full step Rotary Encoder class.

5.2.1 Detailed Description

Three speed full step Rotary Encoder library for Arduino.

[RotaryFullStep.h](#)

Source: <https://github.com/Erriez/ErriezRotaryEncoderFullStep>

Index

read

RotaryFullStep, [10](#)

RotaryFullStep, [9](#)

read, [10](#)

RotaryFullStep, [9](#)

src/RotaryFullStep.cpp, [11](#)

src/RotaryFullStep.h, [11](#)